

Efficient Collective Communication in Distributed Heterogeneous Systems

Prashanth B. Bhat *

Dept. of EE-Systems, EEB 246
University of Southern California
Los Angeles, CA 90089-2562
prabhat@halcyon.usc.edu

C.S. Raghavendra

The Aerospace Corporation
P. O. Box 29257
Los Angeles, CA 90009
raghu@aero.org

Viktor K. Prasanna*

Dept. of EE-Systems, EEB 200C
University of Southern California
Los Angeles, CA 90089-2562
prasanna@ganges.usc.edu

Abstract

The Information Power Grid (IPG) is emerging as an infrastructure that will enable distributed applications – such as video conferencing and distributed interactive simulation – to seamlessly integrate collections of heterogeneous workstations, multiprocessors, and mobile nodes, over heterogeneous wide-area networks. This paper introduces a framework for developing efficient collective communication schedules in such systems. Our framework consists of analytical models of the heterogeneous system, scheduling algorithms for the collective communication pattern, and performance evaluation mechanisms. We show that previous models, which considered node heterogeneity but ignored network heterogeneity, can lead to solutions which are worse than the optimal by an unbounded factor. We then introduce an enhanced communication model, and develop three heuristic algorithms for the broadcast and multicast patterns. The completion time of the schedule is chosen as the performance metric. The heuristic algorithms are FEF (Fastest Edge First), ECEF (Earliest Completing Edge First), and ECEF with look-ahead. For small system sizes, we find the optimal solution using exhaustive search. Our simulation experiments indicate that the performance of our heuristic algorithms is close to optimal. For performance evaluation of larger systems, we have also developed a simple lower bound on the completion time. Our heuristic algorithms achieve significant performance improvements over previous approaches.

1. Introduction

With recent advances in high-speed networks, distributed heterogeneous computing has emerged as an attractive computational paradigm. The Information Power Grid (IPG) [6]

is emerging as an infrastructure that will connect distributed computational sites worldwide. This will create a universal source of computing power, thereby providing pervasive and inexpensive access to advanced computational capabilities. A typical grid-based distributed computing system will consist of a collection of heterogeneous workstations, multiprocessors, and mobile nodes. These nodes communicate with one another using a common set of protocols over different types of communication links, such as ATM, FDDI, Ethernet, and wireless channels. An example of such a system is shown in Figure 1. Such a distributed computing system is heterogeneous both in the computing nodes and in the communication network.

Several research projects, such as Globus [7], Legion [9], and MSHN [13] are developing toolkits and infrastructure support to enable the use of these systems for high performance computing. The issue of data dissemination middleware for wide-area network collaboratories is also being investigated [12]. Our research is a part of the MSHN project [13], which is a collaborative effort between DoD (Naval Postgraduate School), academia (NPS, USC, Purdue University), and industry (NOEMIX). MSHN (Management System for Heterogeneous Networks) is designing and implementing a Resource Management System (RMS) for distributed heterogeneous and shared environments. The goal is to schedule shared compute and network resources among individual applications so that their QoS requirements are satisfied.

The availability of high-speed wide-area networks has also enabled collaborative multimedia applications such as video conferencing, distributed interactive simulation, and collaborative visualization. For example, the FACE project [16] organized world-wide teleconferences among agents in Japan, USA, and the UK. The participating sites in these applications exchange large volumes of multimedia data, such as voice and video. Using the Internet, messages were propagated in about 60 msec between sites in Japan, while it took about 240 msec between Japan and Europe [16].

*Supported by the DARPA/ITO Quorum Program through the Naval Postgraduate School under subcontract number N62271-97-M-0931.

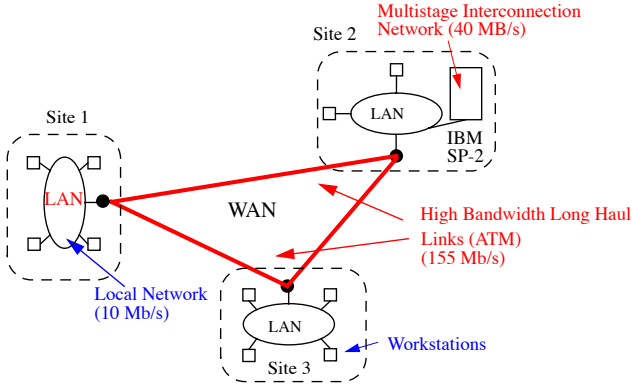


Figure 1. A typical distributed heterogeneous system.

In both of the above scenarios, *viz.*, distributed high performance computing and collaborative multimedia applications, it is extremely important to efficiently perform group communication over a heterogeneous network. Typical group communication patterns are multicast, broadcast, and total exchange. In the multicast pattern, a source node sends the same message to a subset of nodes in the system. The broadcast pattern is a special case of multicast where the message is sent from a source to all the other nodes. In the total exchange communication pattern, every node sends a distinct message to every other node. The goal is to optimize a specified performance measure, *eg.*, minimize the time at which all the messages have been delivered.

In this paper, we develop efficient algorithms for broadcast and multicast in heterogeneous computing environments. These communication patterns occur in several military and commercial applications. In the battlefield, rapid dissemination of work orders and threat scenarios is critical [17]. A global satellite and ground-based networks are used in military battlefield to broadcast messages. The satellite sends the message to a group of base stations as it passes over them. The base stations then co-operatively broadcast the message to the other destinations over ground-based networks. The Internet can also be used to rapidly disseminate important emergency messages. In the past, broadcast and multicast problems have been studied extensively in the context of homogeneous and worm-hole routed networks [10, 14]. Similarly, multicast protocols such as CBT [2], DVMRP, and PIM [5] are now being deployed in wide-area networks. However, these techniques are not appropriate for the distributed network scenarios that we consider in this paper. For example, flooding is a technique where a node simultaneously sends the broadcast message to all its neighbors. The receiving nodes “flood” their neighbors

in turn, until the message is received by all nodes. Some of the nodes could receive the message multiple times, depending on the network topology. Such techniques will not be efficient in wide-area heterogeneous networks, since each point-to-point communication event incurs an additional communication cost. Further, this will also introduce extra network congestion.

Recent research efforts [3] have investigated the problem of efficient broadcast and multicast in a network of heterogeneous workstations. The heterogeneity in the communication capabilities of the workstations was represented by associating a message initiation cost with each workstation. However, heterogeneity in the network was not considered. Based on this communication cost model, heuristic algorithms were developed for the broadcast and multicast problems. The heuristics achieve near-optimal performance for up to 10 nodes. In Section 2, we show that such a communication model can be very ineffective in a system with a heterogeneous network. We give examples where the completion time of a broadcast schedule using such a model is larger than the optimal completion time by an unbounded factor. It is therefore necessary to use a communication framework that considers heterogeneity in both the nodes and network links. Section 3 introduces our new communication model and framework. Our model represents the communication cost between two nodes P_i and P_j using two parameters: (i) a start-up time which accounts for the message initiation cost at P_i , and the network latency from P_i to P_j , and (ii) a data transmission cost which depends on the message size and the bandwidth from P_i to P_j . Using this model, we can consider the distributed system to be a fully connected network with a communication cost C_{ij} between every pair of nodes P_i and P_j . We do not assume a symmetric network, *i.e.*, $C_{ij} \neq C_{ji}$.

Since the problem of finding the optimal broadcast schedule in such a heterogeneous system is NP-complete, we have developed heuristic algorithms based on our communication framework. Our heuristic algorithms produce near optimal solutions for up to 10 nodes when tested with random networks. For larger size systems, it is extremely time consuming to compute the optimal solution. We have therefore developed a lower bound on the completion time. We evaluate the different heuristics by comparing their completion time with the lower bound.

The rest of the paper is organized as follows. In Section 2, we discuss related work and its shortcomings. Section 3 presents our formal model and general framework for collective communication in heterogeneous distributed computing environments. In Section 4, we present several heuristic algorithms for the broadcast and multicast problems. Section 5 compares the performance of our heuristics with previous algorithms, using simulation results. Section 6 identifies future research directions.

2. Shortcomings of Previous Research

Collective communication in homogeneous workstation networks and tightly coupled parallel systems has been thoroughly researched over the years. Communication libraries for frequently used patterns such as *total exchange*, *one-to-all broadcast*, *all-to-all broadcast*, and *gather* have been developed [1, 4, 18, 19].

However, collective communication in heterogeneous systems has not been investigated until very recently [11, 3]. The Efficient Collective Operations (ECO) [11] package was developed for networks of heterogeneous workstations. It implements the same functionality as the collective communication suite in the MPI standard. The ECO approach consists of first partitioning the network into subnets. A subnet consists of hosts which are in the same physical network. The collective communication then proceeds in two phases, inter-subnet and intra-subnet. However, such a two-phase strategy does not always ensure efficient implementations of collective communication patterns. This is especially true if the inter-subnet links are much slower than the intra-subnet links.

Banikazemi *et al.* [3] identified the important problem of performing efficient broadcast and multicast among a cluster of heterogeneous workstations. A homogeneous network was assumed. Their communication model associates a message initiation cost T_i with each of the N workstations. T_i is incurred whenever the i^{th} workstation (P_i) sends a message, independent of the identity of the receiving workstation. Based on this communication cost model, it was shown that broadcast schedules based on binomial trees, which achieve good results in homogeneous systems, can be very ineffective. A $N - 1$ step heuristic algorithm, called Fastest Node First (FNF), was developed. Each step of the heuristic selects a sender and a receiver. The receiver is the node with the lowest T_i among the remaining receivers. The sender is the node that can complete the communication event at the earliest possible time. The FNF heuristic was evaluated for systems with up to 10 nodes [3]. For the examples considered, the completion time of the FNF heuristic was very close to the optimal.

However, there are scenarios where the performance of the FNF heuristic can be sub-optimal. Consider the example where the source has cost 1, there are n nodes with costs $n, n + 1, n + 2, \dots, 2n - 1$, and $2n$ slow nodes with very high costs. In the optimal schedule, the source would first send n messages to nodes with cost $2n - 1, 2n - 2, \dots$, respectively. At time n , the node with cost n has received the message from the source. Immediately after receiving the message, each of these nodes initiates a message to one of the slow nodes. During the time interval $[n, 2n]$, the source sends n more messages to the remaining slow nodes. The schedule completes at time $2n$.

In the FNF schedule, the source will send messages to nodes with cost $n, n + 1, \dots, 2n - 1$ respectively. At time n , n nodes will have received the message. If each node immediately initiates a new message, each of the nodes with costs n to $\frac{3n}{2}$ can reach a slow node by time $2n$. During the time interval $[n, 2n]$, the source sends n more messages to n of the slow nodes. Thus, at time $2n$, $\frac{n}{2}$ of the slow nodes have not yet received the message. The schedule takes $\frac{n}{8}$ extra time units to complete. For large values of n , the completion time of the FNF schedule is much larger than the optimal.

A more significant shortcoming of [3] was the assumption of the homogeneous network. In a typical heterogeneous system, the communication cost depends both on the communication capability of the workstations as well as the network performance. Our paper investigates the impact of heterogeneity in both these aspects. We first illustrate the importance of considering network heterogeneity, using an example. Consider a system with 3 nodes, and pairwise communication costs as shown in Eq (1). The $(i, j)^{th}$ entry of \mathbf{C} ($0 \leq i, j < 3$) denotes the time to send the broadcast message from node P_i to P_j . This includes the message initiation cost on node P_i and also the network latency from P_i to P_j . Section 3 discusses this communication model in detail. Node P_0 is the source.

$$\mathbf{C} = \begin{bmatrix} 0 & 10 & 995 \\ 2000 & 0 & 10 \\ 70 & 5 & 0 \end{bmatrix} \quad (1)$$

To develop a communication schedule based on node heterogeneity alone, we associate a communication cost T_i with each node. This is calculated as the average send cost from node P_i to all the other nodes. Thus, in Eq (1), $T_0 = 335, T_1 = 670, T_2 = 25$. We can now use the FNF heuristic [3] for this problem. Since the heuristic operates on a modified version of the input data (*i.e.*, the average communication costs), we call this the modified FNF heuristic.

For the example of Eq (1), the heuristic begins with node P_0 as the only sender. In the first step, P_2 is selected as the receiver. The communication from P_0 to P_2 takes 995 time units. Both these nodes are ready to send the next message at time 995. In the next step, node P_2 is selected as the sender and P_1 is selected as the receiver. This communication event takes 5 time units. The broadcast therefore takes 1000 time units to complete. Figure 2(a) shows this communication schedule.

However, it is easy to see that the optimal schedule takes only 20 time units. In the first step, P_0 sends a message to P_1 in 10 time units. In the next step, P_1 sends a message to P_2 in 10 time units. This schedule is shown in Figure 2(b). Thus, the use of a single message initiation cost for each node results in a communication schedule which is 50 times worse than the optimal schedule for this example. In the above example, we used the average send cost

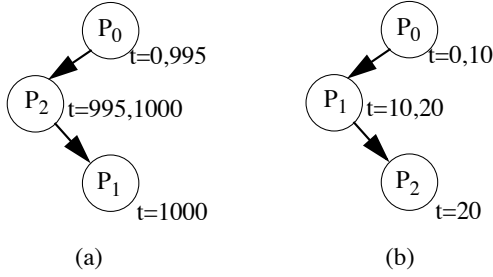


Figure 2. Broadcast schedules for the example in Eq (1): (a) Modified FNF schedule (b) Optimal schedule.

from each sender as its communication cost. Alternatively, we could have used the minimum send cost of each sender as its communication cost T_i . In Eq (1), the costs would then be $T_0 = 10, T_1 = 10, T_2 = 5$. It can be easily verified that the modified FNF heuristic again takes 1000 time units to complete.

The performance of the modified FNF heuristic would be still worse if the value of $C_{2,0}$ was larger. For example, if $C_{2,0}$ was 9995 instead of 995, the completion time would have been 10000 time units, *i.e.* 500 times the optimal completion time. We summarize this observation in the following lemma.

Lemma 1: In the presence of a heterogeneous network, there exist input instances for which the ratio of the completion time of the modified FNF heuristic to the optimal completion time is unbounded. \square

Thus, communication models which consider only node heterogeneity can result in arbitrarily bad performance. It is therefore important to consider both node heterogeneity and network heterogeneity when designing communication algorithms for the broadcast problem.

3. A Communication Framework for Distributed Heterogeneous Systems

We now present our communication scheduling framework for distributed heterogeneous systems. The framework consists of three main components: (a) A communication model, (b) Scheduling heuristics, and (c) Performance metrics. In this section, we describe an enhanced communication model which incorporates node and network heterogeneity. Section 4 describes our heuristic algorithms for broadcast and multicast based on this model. The performance metric used in this paper is the completion time. Other candidate metrics are discussed in Section 7.

3.1. Communication Model

Consider a distributed heterogeneous system (Figure 1) with N nodes. We represent the computing nodes and network links in such a system using a directed graph G with N vertices. An edge (v_i, v_j) in G represents the path between nodes P_i and P_j , which could include links from multiple networks of different latencies and bandwidths. The weight C_{ij} of edge (v_i, v_j) , ($0 \leq i, j < N$) represents the time to send the broadcast message from P_i to P_j . If there exists at least one path between every pair of nodes in the system, G will be a complete graph. The graph is not necessarily symmetric, *i.e.* $C_{ij} \neq C_{ji}$, in general. The information can also be represented as a $N \times N$ communication matrix C , with entries C_{ij} , as shown in Eq (1).

Our communication model represents the network performance between any processor pair (P_i, P_j) using two parameters: a start-up cost T_{ij} and a data transmission rate B_{ij} . The time for sending a m byte message between these nodes is given by $T_{ij} + \frac{m}{B_{ij}}$. A similar communication model has been widely used for tightly-coupled homogeneous distributed memory systems with good results [20]. In networked heterogeneous systems, typical values for the start-up cost could be in the range of 10 to 500 μs , while typical values for the bandwidth could be in the range of kb/s to hundreds of Mb/s. Note that the communication time depends on the identities of both the sender and receiver, unlike previous models [3]. The model thus enables a realistic estimate of the communication time between any pair of nodes.

Table 1 is an example of measured network performance on the GUSTO testbed of the Globus distributed heterogeneous system [7]. The table shows four of the GUSTO sites: NASA AMES, Argonne National Lab, University of Indiana, and USC-ISI. Observe that the network performance varies considerably between different pairs of nodes, and depends on both the source and destination. For instance, the bandwidth between USC-ISI and AMES is much larger than the bandwidth between USC-ISI and IND. Previous communication models [3], which assume that the communication time from node P_i to node P_j is independent of P_j and depends only on the source node P_i , are therefore unlikely to be effective for such systems.

Our model assumes that a node is allowed to simultaneously participate in at most one send and one receive operation. When a node has multiple messages to send, it performs these send operations one after another. Current hardware and software do not easily enable multiple messages to be transmitted simultaneously. Software support for non-blocking and multithreaded communication sometimes allows applications to initiate multiple send and receive operations. However, all these operations are eventually serialized by the single hardware port to the network. Our model

	AMES	ANL	IND	USC-ISI
AMES		34.5/512	89.5/246	12/2044
ANL	34.5/512		20/491	26.5/693
IND	89.5/246	20/491		42.5/311
USC-ISI	12/2044	26.5/693	42.5/311	

Table 1. Latency(ms) / Bandwidth(kbits/s) between 4 GUSTO sites.

accurately represents this phenomenon.

If multiple nodes simultaneously send to any node P_j , we say that node contention occurs at P_j . The model assumes that these messages are received one after the other at P_j . The validity of this assumption can be seen by examining the events involved in a message transmission from P_i to P_j . A control message is first transmitted by P_i . The actual data is sent only after this control message is acknowledged by P_j . If P_j is busy receiving from a different node, it sends the acknowledgement to P_i only after completing the previous receive operation.

Based on the network performance parameters and our communication model, we can calculate the communication time to send the broadcast message between any pair of nodes in the heterogeneous network. This information is used to determine the edge weights of G and the entries of the communication matrix C . The communication matrix for broadcasting a 10 MByte message over the network of Table 1 is shown in Eq (2). Entries are in *sec*.

$$C = \begin{bmatrix} 0 & 156 & 325 & 39 \\ 156 & 0 & 163 & 115 \\ 325 & 163 & 0 & 257 \\ 39 & 115 & 257 & 0 \end{bmatrix} \quad (2)$$

4. Heuristics for Broadcast and Multicast

Consider a communication cost matrix C with N nodes. We first define a lower bound on any communication schedule for the broadcast and multicast problems, and then discuss our heuristic algorithms.

4.1. A Lower Bound

Let P_0 be the source of the broadcast or multicast operation, and D represent the set of destination nodes. $D \subset \{P_1, P_2, \dots, P_{N-1}\}$ for multicast, while $D = \{P_1, P_2, \dots, P_{N-1}\}$ for broadcast. For each node P_i in D , we can compute the shortest path from the source node P_0 to P_i . The weight of this path represents the earliest time at which the broadcast message from P_0 can reach P_i . This is

therefore called the *Earliest Reach Time* of node P_i , denoted as ERT_i .

Lemma 2: A lower bound on any communication schedule for the broadcast or multicast problem is given by

$$LB = \max_{P_i \in D} ERT_i \quad (3)$$

Proof: We know that ERT_i represents the earliest time at which node P_i can be reached. From the definition of the broadcast and multicast communication pattern, the message must reach every node in D . Hence, no communication schedule can complete until the node with the maximum ERT is reached. Eq (3) therefore gives a lower bound on the completion time. \square

The lower bound is not tight, since it assumes that the messages from the source to each destination can proceed in parallel. Thus, the optimal completion time could be significantly larger than the lower bound.

Lemma 3: For any instance of the multicast or broadcast problem, the optimal completion time is bounded by $|D| \times LB$, i.e.,

$$\frac{OptimalCompletionTime}{LB} \leq |D| \quad (4)$$

Further, this ratio is tight.

Proof: The lower bound LB of Eq (3) is the communication time to send the message from the source to the farthest node. Thus, the communication time to send a message from the source to any node is $\leq LB$. We can always construct a communication schedule in which the source sequentially sends $|D|$ messages to all the destinations. The $|D|$ communication steps can therefore be completed in at most $LB \times |D|$ time units.

To prove that the ratio is tight, consider the broadcast problem on the communication cost matrix of Eq (5). In this matrix, $C_{0j} = 10, (0 < j < N)$. Also, $C_{ij} = 100, (0 < i < N, i \neq j)$. The diagonal entries $C_{ii} = 0, (0 \leq i < N)$. The shortest path to every node P_i is the direct path (P_0, P_i) . The lower bound would be the maximum outgoing edge from P_0 , i.e., 10. However, the optimal schedule has a completion time of $10 |D|$. Thus, there exist examples wherein the optimal completion time is $|D|$ times as large as our simple lower bound. \square

$$C = \begin{bmatrix} 0 & 10 & 10 & \dots & 10 \\ 100 & 0 & 100 & \dots & 100 \\ \dots & \dots & \dots & \dots & \dots \\ 100 & 100 & 100 & \dots & 0 \end{bmatrix} \quad (5)$$

4.2. Computing the Optimal Schedule

The possible number of communication schedules for a broadcast or multicast problem instance with N nodes is exponential in N . The completion times of these schedules can

vary considerably, depending on the performance of the heterogeneous network links. Finding the optimal communication schedule is an NP-Complete problem. However, for systems with a small number of nodes, we can find the optimal schedule using exhaustive search. Our algorithm, which uses a branch-and-bound strategy, computes the optimal solution for up to 10 nodes in a reasonable amount of time. For small system sizes, we shall compare the performance of our heuristic algorithms with the optimal solution.

4.3. Our Heuristic Algorithms

Our algorithms for the broadcast and multicast problems can be described using the following formalism. The nodes are partitioned into three sets, **A**, **B**, and **I**. At any time, set **A** consists of nodes which have already received the message. Set **B** consists of nodes which must receive the message in the future. **I** contains the other nodes. Initially, set **A** consists of the source node while set **B** consists of the destination nodes for the multicast, *i.e.* $\mathbf{B}=\mathbf{D}$. For the broadcast problem, $\mathbf{I} = \phi$.

At every step, a sender from **A** and a receiver from **B** are chosen. For the multicast problem, the message could also be relayed through one of the nodes in **I**, if this path incurs lower communication time. After each communication event, the receiver node (and the intermediate node, if one was chosen) is moved to **A**. The communication schedule involves $|\mathbf{D}|$ such steps. We now present the baseline algorithm and our FEF, ECEF, and look-ahead heuristics.

Baseline Algorithm

We use the modified FNF heuristic [3] as a baseline algorithm. This algorithm associates a single communication cost with each node rather than a distinct cost for each pair of nodes. We use the average send cost from node P_i to all the other nodes as its communication cost T_i . The FNF heuristic algorithm [3] consists of $N - 1$ steps. At every communication step, the node from **B** with the lowest T_j is chosen as the receiver. A sender is chosen such that the communication event can be completed at the earliest possible time. This is the node P_i which has the minimum value of

$$R_i + T_i \quad (6)$$

where R_i is the ready time of the sender P_i .

Fastest Edge First (FEF):

Each step of our FEF heuristic selects the smallest weight edge (i, j) where P_i belongs to **A** and P_j belongs to **B**. The choice of the edge determines both the sender and receiver node for the corresponding communication step. P_j is then moved from **B** to **A**. The communication step starts at R_i ,

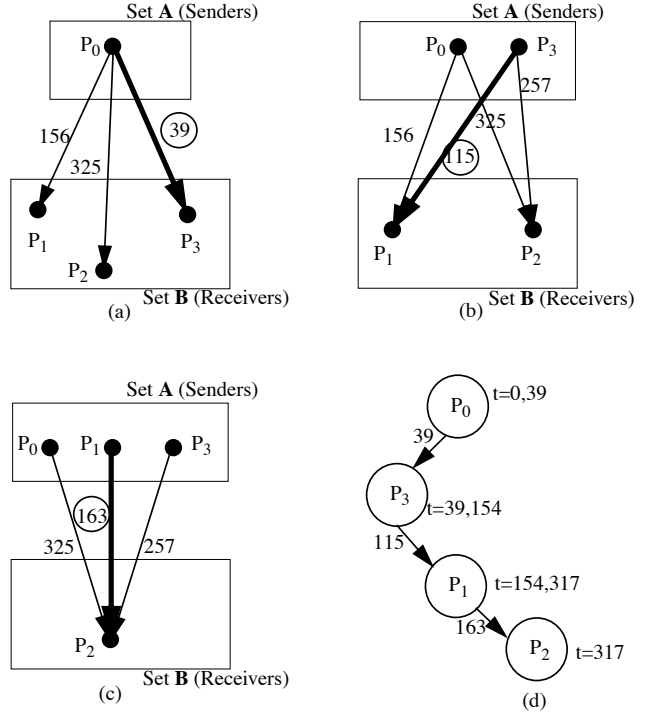


Figure 3. FEF communication schedule for the 4 node example of Eq (2).

and takes $C_{i,j}$ time units. During this time, both P_i and P_j are busy.

The algorithm initially sorts the outgoing edges from each node in increasing order of their weights. This phase takes $O(N^2 \log N)$ time. The senders in **A** are then sorted in increasing order of their minimum weight outgoing edge. The new node added to set **A** at every step is inserted into the sorted sender list based on its minimum weight outgoing edge. The algorithm terminates after all the destination nodes have been moved to **A**. This involves $N - 1$ steps for the broadcast algorithm and a maximum of $N - 1$ steps for the multicast algorithm. The running time for this phase is also $O(N^2 \log N)$. The overall running time of the FEF heuristic is therefore $O(N^2 \log N)$.

Figure 3 shows the steps in the FEF heuristic for the broadcast problem in the 4 node system of Eq (2). Figure 3(a) shows the initial situation when set **A** contains only the source node, and set **B** contains the other nodes. The figures show the edge weights of only the edges in the **A-B** cut. Figures 3(b)-3(c) show the sequence in which the FEF heuristic moves edges from **B** to **A**. Figure 3(d) shows the broadcast tree for this schedule.

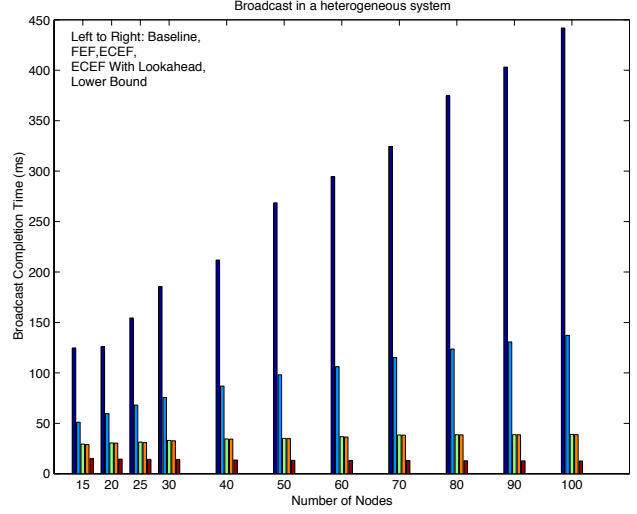
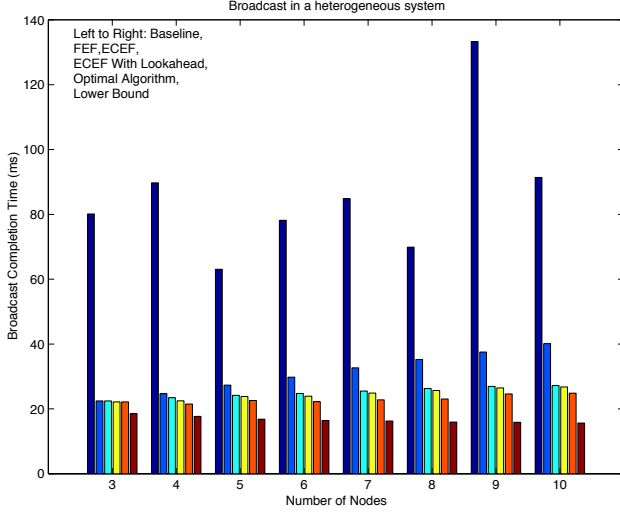


Figure 4. Simulation results for broadcast in a heterogeneous system.

Earliest Completing Edge First (ECEP):

The structure of our ECEP heuristic algorithm is similar to the FEF heuristic. At every step, an edge (i, j) is selected, where P_i belongs to \mathbf{A} and P_j belongs to \mathbf{B} . The choice of the edge considers both the weight of the edge and the ready time of the sender. The chosen communication event is the one that can complete earliest. Thus, the chosen edge is the one that minimizes the sum

$$R_i + C_{i,j} \quad (7)$$

over all senders P_i and receivers P_j , where R_i is the ready time of sender P_i . As in the FEF heuristic, a sorted list of senders is maintained. The senders are sorted based on both their ready time and their minimum weight outgoing edge. The heuristic has a running time of $O(N^2 \log N)$.

Look-ahead Algorithm:

Our look-ahead algorithm is an enhanced version of the ECEP heuristic. At each step of the heuristic, a *look-ahead value* L_j is calculated for each node P_j in \mathbf{B} . This value quantifies the “goodness” of moving node P_j from \mathbf{B} to \mathbf{A} . At each step, the algorithm first computes the value of L_j for all nodes in \mathbf{B} . As in the ECEP heuristic, an edge is then selected from the \mathbf{A} - \mathbf{B} cut. The chosen edge is the one that minimizes the sum

$$R_i + C_{i,j} + L_j \quad (8)$$

The look-ahead function can be defined in several ways. We have used the following look-ahead measure.

$$L_j = \min_{P_k \in \mathbf{B}} C_{j,k} \quad (9)$$

Thus, for a given node P_j in \mathbf{B} , the minimum communication cost from itself to all the other nodes in \mathbf{B} is used as the look-ahead value. Intuitively, such a look-ahead function increases the usefulness of P_j as a sender, if it is moved to \mathbf{A} .

The running time of the look-ahead algorithm is $O(N^3)$, since the evaluation of the look-ahead measure for each element of \mathbf{B} at every step takes $O(N)$. Alternative look-ahead functions can also be used, such as the average of the communication costs from P_j to other nodes in \mathbf{B} . L_j could also be calculated as the average cost of senders to receivers, assuming that P_j is made a sender. This look-ahead function has a computational complexity of $O(N^2)$, and the overall running time will therefore be $O(N^4)$. Our experiments in Section 5 use the look-ahead measure of Eq (9).

5. Experimental Results

We have developed a software simulator that executes the heuristic algorithms of Section 4, and calculates the completion time for each of them. The inputs to the simulator are the number of nodes, the size of the message to be broadcast or multicast, and the range of start-up times and bandwidths in the heterogeneous network. The simulator generates a random communication matrix based on these parameters. For the case of multicast, the number of destinations is given as input, and the simulator randomly chooses destination nodes. The simulator then executes the heuristic algorithms on 1000 random input configurations and reports the average completion times.

Figure 4 compares the performance of the different communication scheduling heuristics for the broadcast problem with a message size of 1 MB. The pairwise network laten-

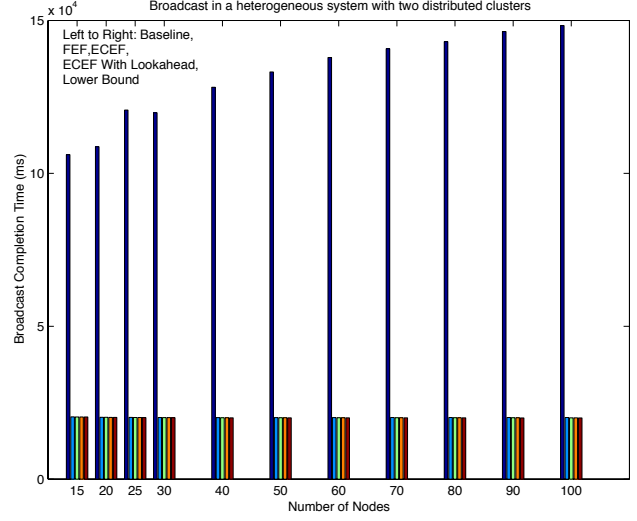
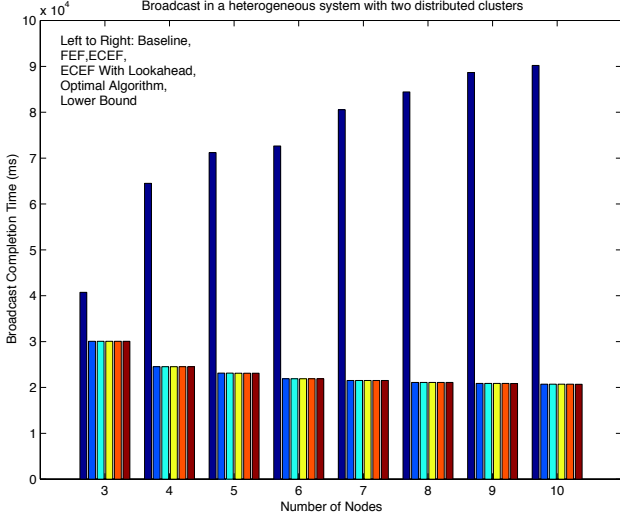


Figure 5. Simulation results for broadcast in a heterogeneous system with 2 distributed clusters.

cies and bandwidths are chosen in the ranges of $10 \mu sec$ to $1 msec$, and $10kB/s$ to $200MB/s$ respectively. The graph shows the completion time for the baseline algorithm, the FEF, ECEF, and look-ahead heuristics, and our simple lower bound. For small system sizes (upto 10 nodes), the optimal completion time is also shown. Since our lower bound is not tight, it is typically much lesser than the optimal completion time. The graph shows that the completion time of our heuristic algorithms is always close to the optimal. The ECEF and look-ahead algorithms have a lower completion time than that of the FEF heuristic. The completion time of the baseline algorithm is significantly larger than that of the other heuristics. This shows the benefit of using a communication model which accurately represents heterogeneity in the network, as well as in the nodes.

The performance advantage of our heuristic algorithms over the baseline algorithm can also be seen in Figure 5. Figure 5 considers a system with two distinct geographically distributed clusters. It is assumed that half the nodes are in the first cluster, while the other nodes are in the second cluster. The heterogeneous network is assumed to be fast within each cluster, but is slow across clusters. For the intra-cluster networks, the latencies and bandwidths are in the ranges of $10 \mu sec$ to $1 msec$, and $10MB/s$ to $200MB/s$ respectively. For the inter-cluster networks, the latencies and bandwidths are in the ranges of $1msec$ to $20msec$, and $10kB/s$ to $50kB/s$ respectively. As before, the size of the broadcast message is 1 MB.

Figure 6 shows the completion time for multicast in a 100 node system. The number of multicast destinations is increased from 5 to 90. For the case of k destinations, 1000 experiments are performed with k randomly chosen destinations. The average completion time is plotted in Figure 6.

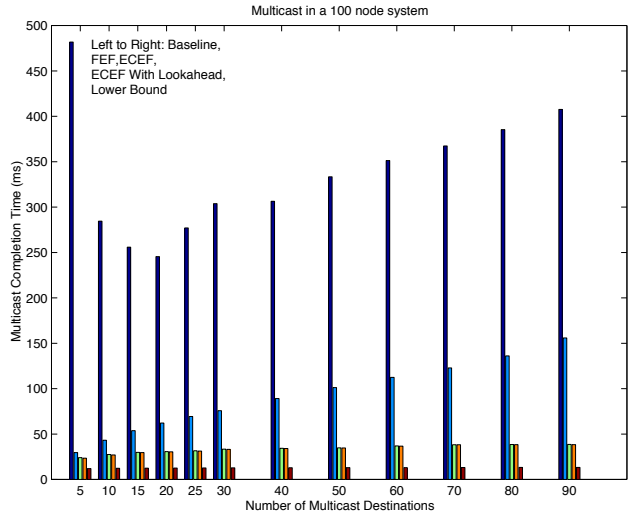


Figure 6. Simulation results for multicast.

Observe that the heuristic algorithms again significantly outperform the baseline algorithm.

6. Research Issues

The experimental results of Section 5 clearly show the performance benefits of our heuristic algorithms. However, there are scenarios in which some of our heuristics can have poor performance. Consider the asymmetric communication cost matrix of Eq (10), which could be a system with Asymmetric Digital Subscriber Lines (ADSL).

$$\mathbf{C} = \begin{bmatrix} 0 & 2 & 2 & 2.1 & 2 \\ 100 & 0 & 100 & 100 & 100 \\ 100 & 100 & 0 & 100 & 100 \\ 1000 & 0.1 & 0.1 & 0 & 0.1 \\ 100 & 100 & 100 & 100 & 0 \end{bmatrix} \quad (10)$$

In the optimal broadcast schedule, P_0 sends the message to P_3 in step 1, and then P_3 then sends messages to the other nodes in steps 2, 3, and 4. This has a completion time of 2.4 time units. However, the ECEF heuristic sends the message from P_0 to P_1 in step 1, P_0 to P_2 in step 2, P_0 to P_4 in step 3, and P_0 to P_3 in step 4. The completion time is 8.4 time units. The look-ahead algorithm does find the optimal schedule. It chooses the node P_3 as the receiver in the first step, since P_3 has a low-cost outgoing edge.

However, the performance of the look-ahead schedule is poor for the communication matrix of Eq (11). The algorithm takes 4.1 time units (P_0 to P_1 , P_1 to P_2 , P_2 to P_3 , and P_3 to P_4). The optimal schedule takes only $2.2 + 2\epsilon$ time units (P_0 to P_3 , P_3 to P_4 , P_4 to P_1 , P_4 to P_2). For larger systems, the difference between the completion times of the look-ahead and optimal schedules can be much higher.

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 1 & 1.1 & 100 \\ 100 & 0 & 1 & 100 & 100 \\ 100 & 100 & 0 & 1 & 100 \\ 100 & 100 & 100 & 0 & 1.1 \\ \epsilon & \epsilon & \epsilon & \epsilon & 0 \end{bmatrix} \quad (11)$$

However, communication matrices such as Eq (11) do not typically occur in real scenarios. Often, \mathbf{C} is symmetric. The triangle inequality is also usually valid, i.e.,

$$C_{ij} \leq C_{ik} + C_{kj}, 0 \leq k < N \quad (12)$$

For such a system, stronger performance bounds than Eq (4) could be shown. We are investigating this issue.

We are also investigating new heuristic schedules based on the Minimum Spanning Tree (MST) and Steiner Tree algorithms. The steps in our FEF algorithm are identical to Prim's MST algorithm. We are currently investigating a *progressive MST* approach. This is an enhancement to Prim's algorithm which accounts for the ready time of each node. After each step of the algorithm, some of the edge weights are updated to reflect the change in ready times. We are also investigating a *two-phase* approach. During the first phase, a MST is constructed. The structure of the MST is used to guide the selection of intermediate nodes for the second phase, which constructs the heuristic schedule.

The main difference between the MST problem and our broadcast problem is the cost metric. The metric in the MST problems is usually the total weight of edges

in the spanning tree. In contrast, the completion time of the broadcast and multicast problems is the time at which all nodes have received the message. Delay-constrained MST problems, which minimize the maximum delay between the source and any destination, have also been considered [15]. However, this metric is also different from the completion time. Consider the example of Eq (10). The delay-constrained algorithm would create a MST with edges (P_0, P_1) , (P_0, P_2) , (P_0, P_3) , and (P_0, P_4) . Although the maximum delay is 2.1, the completion time is 8.1 time units. In fact, if the triangle inequality of Eq (12) holds, the delay-constrained algorithm will always send $|D|$ messages sequentially from the source to each destination.

A second difference is that the widely known MST algorithms of Prim and Kruskal were developed for undirected graphs. Our progressive and two-phase techniques can build upon these techniques if the heterogeneous network is symmetric. For asymmetric networks, MST algorithms for directed graphs can be used [8].

In designing a heuristic, we must give special attention to two kinds of nodes: (a) Nodes which are hard to reach from every other node, and are also unable to reach other nodes quickly. The message to such a node should be sent early in the schedule, so that this communication event does not delay the completion time. (b) Nodes which are a little hard to reach, but which can reach many other nodes very easily. Such nodes should be selected early, so that they can relay the message to the other nodes.

We are therefore exploring an alternating *near-far* approach. All nodes are initially sorted in increasing order of their *ERT*. In the first two steps, messages are sent to the nearest node (say P_i), and to the farthest node (say P_j). From this point onwards, P_i and its recipients will send messages to the *near* nodes. This group always selects the nearest unreachable node at every step. P_j and its recipients will send messages to the *far* nodes. This group selects the farthest unreachable node. Such a near-far strategy is likely to balance the two conflicting goals discussed above.

For the multicast problem, we shall enhance our algorithm to relay messages through nodes in the intermediate set I , defined in Section 4.3. Our current algorithm does not incorporate this aspect. The problem of scheduling multiple simultaneous multicasts will also be considered.

The previous sections have illustrated the use of our framework for a specific cost model and performance metric. We now discuss some variations and extensions of these components. Our communication model assumed that a node can send and receive at most one message at any time. In a *non-blocking* communication model, this assumption is relaxed. After an initial start-up time, the sender can initiate a new message. The first message is completed by the network without further intervention by the sender. Thus, a node could send out several messages before the first mes-

sage reaches the receiver. Similar assumptions can be made at the receiver too.

We have used the completion time as our performance metric. *Robustness* metrics can be used to measure the ability of a communication schedule to reach all destinations, in spite of intermediate node or link failures. A communication schedule could increase its robustness measure by sending redundant messages for fault tolerance. Alternatively, acknowledgement schemes and time-out parameters could be used to detect failures before resending a message over a different path. Another candidate metric is the *amount of transmitted data*.

7. Conclusion

Efficient communication support is extremely important for several distributed computing scenarios, such as collaborative multimedia applications and parallel high performance computing over the IPG. This paper has introduced an analytical framework for designing efficient collective communication algorithms. The main components of our framework are a communication model to represent the heterogeneous network and nodes, performance metrics, and scheduling algorithms. Based on this framework, we have developed efficient solutions for broadcast and multicast. We have also identified several promising research directions to extend our work. We believe that future work along these directions can accelerate the widespread use of distributed heterogeneous computing.

Acknowledgment

We thank Mr. M. Banikazemi and Dr. D.K. Panda of the Ohio State University for providing us access to their program for computing the optimal communication cost in a system with heterogeneous nodes. Our branch-and-bound program to compute the optimal schedule in a system with a heterogeneous network uses some of their subroutines.

References

- [1] V. Bala, J. Bruck, R. Cypher, P. Elustondo, A. Ho, C.-T. Ho, S. Kipnis, and M. Snir. CCL: A portable and tunable collective communication library for scalable parallel computers. *IEEE Trans. Parallel and Distributed Systems*, 6(2):154–164, February 1995.
- [2] T. Ballardie, P. Francis, and J. Crowcraft. Core Based Trees (CBT) – an architecture for scalable inter-domain multicast routing. In *Proc. ACM SIGCOMM*, pages 85 – 95, 1993.
- [3] M. Banikazemi, V. Moorthy, and D. K. Panda. Efficient collective communication on heterogeneous networks of workstations. In *Proc. Intl. Conf. Parallel Processing*, pages 460–467, 1998.
- [4] J. Bruck, D. Dolev, C.-T. Ho, M.-C. Rosu, and R. Strong. Efficient Message Passing Interface (MPI) for parallel computing on clusters of workstations. *Journal of Parallel and Distributed Computing*, 40(1):19–34, January 1997.
- [5] S. E. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. G. Liu, and L. Wei. An architecture for wide-area multicast routing. In *Proc. ACM SIGCOMM*, 1994.
- [6] I. Foster and C. Kesselman, Eds. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1998.
- [7] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *Intl. Journal of Supercomputer Applications*, 11(2):115–128, 1997.
- [8] H. N. Gabow, Z. Galil, T. Spencer, and R. E. Tarjan. Efficient algorithms for finding Minimum Spanning Trees in undirected and directed graphs. *Combinatorica*, 6(2):109–122, 1986.
- [9] A. S. Grimshaw and W. A. Wulf. Legion – a view from 50,000 feet. In *Proc. Fifth IEEE Intl. Symp. on High Performance Distributed Computing*, August 1996.
- [10] X. Lin, P. K. McKinley, and L. M. Ni. Performance evaluation of multicast wormhole routing in 2D-mesh multicomputers. In *Proc. Intl. Conf. Parallel Processing, Vol. I*, pages 435–442, August 1991.
- [11] B. B. Lowekamp and A. Beguelin. ECO: Efficient Collective Operations for communication on heterogeneous networks. In *Proc. 10th Intl. Parallel Processing Symposium*, pages 399–405, April 1996.
- [12] G. R. Malan, F. Jahanian, and P. Knoop. Comparison of two middleware data dissemination services in a wide-area distributed system. In *Proc. Intl. Conf. Distributed Computing Systems*, May 1997.
- [13] MSHN Web Page. <http://www.mshn.org>.
- [14] D. K. Panda. Issues in designing efficient and practical algorithms for collective communication on wormhole-routed systems. In *ICPP Workshop on Challenges for Parallel Processing*, pages 8–15, August 1995.
- [15] H. F. Salama, D. S. Reeves, and Y. Viniotis. The Delay-Constrained Minimum Spanning Tree problem. In *Proc. IEEE Symposium on Computers and Communications*, pages 699–703, July 1997.
- [16] T. Tachikawa, H. Higaki, and M. Takizawa. Group communication protocol for realtime applications. In *Proc. 18th IEEE Intl. Conf. Distributed Computing Systems*, pages 40–47, May 1998.
- [17] M. Tan, M. D. Theys, H. J. Siegel, N. B. Beck, and M. Jurczyk. A mathematical model, heuristic, and simulation study for a basic data staging problem in a heterogeneous networking environment. In *Proc. Heterogeneous Computing Workshop*, pages 115–129, March 1998.
- [18] R. Thakur and A. Choudhary. All-to-all communication on meshes with wormhole routing. In *Proc. 8th Intl. Parallel Processing Symposium*, pages 561–565, April 1994.
- [19] K. Verstoep, K. Langendoen, and H. Bal. Efficient reliable multicast on Myrinet. In *Proc. Intl. Conf. Parallel Processing*, volume III, pages 156–165, August 1996.
- [20] C.-L. Wang, P. B. Bhat, and V. K. Prasanna. High-performance computing for vision. *Proceedings of the IEEE*, 84(7):931–946, July 1996.